

TEC-0084

# Three-Dimensional Modeling Station, Phase II

Lei Zhang  
Javier F. Maida  
Gregory D. James

Mandex, Inc.  
12500 Fair Lakes Circle, Suite 300  
Fairfax, VA 22033-3804

October 1996

19961029 102

Approved for public release; distribution is unlimited.

Prepared for:  
U.S. Army Corps of Engineers  
Topographic Engineering Center  
7701 Telegraph Road  
Alexandria, Virginia 22315-3864

DTIC QUALITY INSPECTED 1



US Army Corps  
of Engineers  
Topographic  
Engineering Center

T

E

C



**Destroy this report when no longer needed.  
Do not return it to the originator.**

---

**The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.**

---

**The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.**

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 1996	3. REPORT TYPE AND DATES COVERED Technical October 1993 - December 1995
4. TITLE AND SUBTITLE  Three-Dimensional Modeling Station, Phase II			5. FUNDING NUMBERS  DACA76-93-C-0027
6. AUTHOR(S)  Lei Zhang, Javier F. Maida, Gregory D. James			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Mandex, Inc. 12500 Fair Lakes Circle, Suite 300 Fairfax, VA 22033-3804			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  U.S. Army Topographic Engineering Center 7701 Telegraph Road Alexandria, VA 22315-3864			19. SPONSORING / MONITORING AGENCY REPORT NUMBER  TEC-0084
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION / AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) The objective of this Phase II SBIR project was to provide the U.S. Army Topographic Engineering Center (TEC) with a Three-Dimensional (3-D) Modeling Station composed of hardware, software, and development tools, and to create and populate a data base with models of suitable features found in Defense Mapping Agency's Interim Terrain Data (ITD) set, at an appropriate level of detail. A low-end DOS-based system and a high-end UNIX-based system were configured to meet the varied requirements of simulator data base development, and over 160 3-D feature models were created using the modeling stations. These models represent terrain features, both natural and man-made, defined in the DMA's ITD. Models are available in AutoCAD DXF, IRIS Open Inventor (IV), and Wavefront OBJ formats. A relational data base was designed and created to store the model information, and serves as a standard model library. A DBMS system with the X-Window/Motif graphical user interface was developed to provide tools for model data manipulation and management. Dynamic model placement tools were developed for placing models into simulation scenes. Using these tools, the following three feature types can be placed: point, linear, and area.			
14. SUBJECT TERMS Model Library, 3-D Modeling, Model Data Set, Computer Graphics, Graphics Workstation, Simulation			15. NUMBER OF PAGES 54
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UNLIMITED

# TABLE OF CONTENTS

<b>LIST OF FIGURES</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>vi</b>
<b>PREFACE</b>	<b>vii</b>
<b>1. INTRODUCTION</b>	<b>1</b>
<b>2. DOS-BASED MODELING STATION</b>	<b>3</b>
<b>2.1 Hardware</b>	<b>3</b>
2.1.1 Computer Workstation	3
2.1.2 Graphics Display	3
2.1.3 Digitizer	3
2.1.4 Scanner	3
<b>2.2 Software</b>	<b>4</b>
2.2.1 Modeling Software	4
2.2.2 Utility Software	4
2.2.3 Directories for User Data	5
2.2.4 Operating System	6
<b>3. UNIX-BASED MODELING STATION</b>	<b>7</b>
<b>3.1 Hardware</b>	<b>7</b>
3.1.1 Computer Workstation	7
<b>3.2 Software</b>	<b>7</b>
3.2.1 Visualization and Development Software	7
3.2.2 Data Directories	8
3.2.3 MINFO DBMS Software	9
<b>4. 3D TERRAIN FEATURE MODELS</b>	<b>10</b>
<b>4.1 Model Category</b>	<b>10</b>
<b>4.2 Texture Maps</b>	<b>10</b>
<b>4.3 Texture Mapping and Format Conversion</b>	<b>11</b>
4.3.1 IRIS Showcase	11
4.3.2 Ez3d 1.0	13
<b>5. 3D MODEL DATABASE</b>	<b>17</b>
<b>5.1 Database Structure</b>	<b>17</b>
<b>5.2 Database Backup and Recovery Procedure</b>	<b>17</b>
<b>5.3 DBMS Software</b>	<b>18</b>

5.3.1 Preparing to Use the Software	18
5.3.2 User Guide	19
5.4 Database Management Using INFORMIX-SQL	23
<b>6. DYNAMIC MODEL PLACEMENT TOOLS</b>	<b>24</b>
6.1 Input Specification File	25
6.2 Output Specification File	25
6.3 Model Placement Algorithms	26
6.3.1 Point feature	26
6.3.2 Linear feature	26
6.3.3 Area feature	28
6.4 How to Run the Tool	29
<b>7. REFERENCES</b>	<b>30</b>
 <b>APPENDIX A ITD FEATURE MODEL LISTING</b>	 <b>31</b>
A-1. Obstacles	31
A-2. Slop/Surface Configurations	33
A-3. Soil/Surface Materials	33
A-4. Surface Drainage	33
A-5. Transportation	35
A-6. Vegetation	38
 <b>APPENDIX B DATABASE TABLES AND DATA ELEMENTS</b>	 <b>41</b>
B-1. model_info	41
B-2. country	42
B-3. service	42
B-4. branch	42
B-5. category	43
B-6. distribution	43
 <b>APPENDIX C AN EXAMPLE OF USING Ez3d TO ADD TEXTURE</b>	 <b>44</b>

## LIST OF FIGURES

FIGURE		PAGE
6-1	Interface between DrawLand and the model placement tools	24
6-2	Two scenarios that might occur at the corner of two line segments	27
6-3	Boundary crossing points 1, 2, 3 and 4	28

## LIST OF TABLES

TABLE		PAGE
5-2	IRIS Performer supported data formats	20
6-1	Format of the input description file	25
6-2	Format of the output description file	26

## **PREFACE**

This report was prepared under contract DACA76-93-C-0027 for the U.S. Army Topographic Engineering Center, Alexandria, Virginia 22315-3864 by MANDEX, Inc., Fairfax, Virginia 22033. The Contracting Officer's Representative was Mr. Vineet Gupta.



## 1. INTRODUCTION

Computer Graphics Simulation is widely used by the U.S. military services, especially for training, tactical planning, system design, concept evaluation and testing. These simulations allow trainees to develop their understanding of a tactical scenario or improve their skill level with a weapon system by becoming familiar with important aspects of its operation and function. Simulations allow this to be done without incurring the costs of wear and tear, depreciation or loss of military assets, consumption of supplies (fuel, ammunition, targets, etc.), amortization of costly facilities (e.g. a test range), or the risk of injury to the trainee and others. Furthermore, the training process can be accelerated by numerous repetitions of important or new sequences, while spending less time on skills already learned.

While computer graphics simulation can achieve large cost savings for the military, its potential is limited by the degree of realism that is achievable and by the offsetting costs and time involved in developing three dimensional data sets. Major limitations on realism are: the visual detail with which real environments and objects can be depicted in a computer image; and the speed at which the image can be changed, simulating motion of the objects in either the image and/or the viewer.

Currently, hardware developments are rapidly increasing the speed and capacity of computers to represent visual data and to modify images. Most graphics software tools are also seeing an improvement, and hardware and software prices continue to decline. These trends are supporting improvements in the quality of computer graphics in many applications, as well as stimulating new types of applications. By contrast, however, development of three dimensional graphics data for simulation purposes is severely impeded by the cost of developing accurate and realistic images of terrain, architectural and engineering structures, vehicles, military equipment, etc. Thus, the broad use of simulators is limited by the lack of numerous detailed and accurate model data sets.

The Battlefield Visualization Division at TEC is tasked with researching ways to improve and apply computer image technology in support of a variety of military tasks. To do this, they have developed and configured hardware and software systems with specific functions designed to accomplish various components of the overall computer simulation database development process. Among the missing components of this process that have been recognized by TEC are the needs for a standard library of feature data that can be used to populate simulation scenes and for a standard hardware configuration, coupled with standard software and data interchange formats.

To meet these needs, MANDEX, Inc. was awarded a Phase I SBIR, in 1992, entitled "Three Dimensional Modeling Station," to study modeling capabilities currently in use in the visual simulation community, and to examine the requirements for developing a standard library of simulator models. As a result of the Phase I work, recommendations were made to procure and integrate two hardware and software workstation configurations, and to develop a set of modeling tools to create, import, and maintain a model library of features like those found in the Defense Mapping Agency's (DMA) Interim Terrain Data (ITD). A high-end system and a low-end system were recommended to meet the varied requirements of simulator database

development. Simple modeling efforts can be performed efficiently on the DOS-based system while leaving the more powerful UNIX system for complex, computation intensive tasks<sup>[1]</sup>.

As a result of the work performed in Phase I, MANDEX, Inc. was also awarded a Phase II effort to provide the U.S. Army Topographic Engineering Center's (TEC) Battlefield Visualization Division, with a Three-Dimensional Modeling Station, composed of hardware, software, and development tools to create and populate a database with models of suitable features found in DMA's Interim Terrain Data, at appropriate levels of detail.

To accomplish this, MANDEX implemented the recommendations from our Phase I work in a two part effort. During year one of this Phase II project, a low-end, DOS-based computer system (486/50) was configured and loaded with AutoCAD Release 12 software from AutoDesk, Inc., and MicroStation software developed by Intergraph. Over 160 three-dimensional feature models were created using the DOS-based modeling station. These models represent terrain features, both natural and man-made, defined in the DMA's Interim Terrain Data and are of significance to tactical military operations. The DOS-based modeling station and the feature models saved in AutoDesk's Data Exchange Format (DXF) were delivered to TEC at the end of year one.

In year two of this Phase II project, a high-end, UNIX-based computer system (SGI Indigo 2) was configured and loaded with AutoCAD UNIX version, IRIS Performer visualization software, and Informix database software. Three-dimensional feature models and modeling tools were also migrated from the PC to the UNIX workstation. A relational database was designed and created to store model information. A database management system and associated graphical user interface (GUI) were developed to provide easy access to the model information. In addition to the models created in year one, a set of textured terrain feature models were created and loaded into the model database. Dynamic model placement tools were also developed to perform point, linear, and area feature placements according to operator specifications. These tools will be integrated into the DrawLand software currently being developed by TEC<sup>[2]</sup>.

This final report for the SBIR Program Phase II effort entitled, "Three Dimensional Modeling Station Phase II," presents a detail of the work performed by MANDEX, Inc. for TEC's Battlefield Visualization Division, Alexandria, Virginia.

The report is organized into seven sections. The modeling capabilities, system hardware and software configurations of the DOS-based and the UNIX-based modeling stations are discussed in Sections 2 and 3, respectively. Section 4 documents the three-dimensional feature models created under the Phase II effort, and discusses issues related to texture mapping and model format conversion. Model database structure and DBMS software development are in Section 5. In Section 6, the dynamic model construction and placement tools are discussed in detail. Section 7 gives a list of references.

## **2. DOS-BASED MODELING STATION**

### **2.1 Hardware**

#### **2.1.1 Computer Workstation**

The workstation combined with the appropriate software (Section 2.2) is used as an entry-level tool for the development of three-dimensional Interim Terrain Data (ITD) computer models at low and medium levels of detail. Unfortunately, the computer's processing power is not sufficient for the creation of high level detail models, and the system rendering capabilities are limited to static objects.

The computer system consists of:

- (a) Intel 80486DX2/66 MHz Processor
- (b) Micronics EISA/VESA Local Bus Motherboard
- (c) 16 Megabytes of Main Memory
- (d) 1.2 Gigabytes of Storage
- (e) Adaptec AHA-2742 EISA SCSI-2 Controller
- (f) Internal NEC 3XI Multispin CD-ROM Reader
- (g) Internal 4 GB Wang DAT Tape Drive
- (h) 5¼" and 3½" TEAC Floppy Drives
- (i) Omnikey 101 Keyboard
- (j) Torpedo 9000 Local Bus Graphics Card

Note: Item (j) has been replaced by an ATI SVGA Local Bus Graphics Card

#### **2.1.2 Graphics Display**

The MAG MX17F 17-inch color monitor is used to display three-dimensional wireframe models and static renderings at various resolutions. The graphics display provides digital controls for horizontal and vertical screen settings. It also provides memory capabilities for user selectable resolution modes. However, the monitor's LED readout does not always state the correct resolution.

#### **2.1.3 Digitizer**

The GTCO Roll-up with 16-button cursor digitizing tablet is used to input coordinate information and to select menu items in the modeling software application. The tablet can be easily rolled-up and transported, and has very high resolution and accuracy. Because it is soft, sharp objects can damage its surface.

#### **2.1.4 Scanner**

The Envisions 800 DPI color scanner is used to obtain textures and other images from hard copy sources. It is capable of obtaining high quality images at various resolutions but requires a driver

to interface with the operating system and the application software used to capture and display the scanned images.

## **2.2 Software**

### **2.2.1 Modeling Software**

The modeling application software is used to create three-dimensional wireframe model representations of ITD features.

#### **(a) AutoCAD Release 12**

AutoCAD software provides a suite of tools for efficiently creating three-dimensional models. AutoCAD Data Interchange Format (DXF) is used among industry applications as a standard for sharing information. Unfortunately, AutoCAD native DWG and DXF file formats do not support texture mapping.

To execute AutoCAD, type **acadr12** at the DOS prompt.

#### **(b) Microstation Version 5**

Microstation provides functionality similar to AutoCAD. In addition, it supports texture mapping but does not offer as wide a selection of third-party application programs as AutoCAD.

To execute Microstation, type **ustation** at the DOS prompt.

### **2.2.2 Utility Software**

Utility software is used by the system to gain access to the hardware devices and to the functionality they provide.

#### **(a) SCSI Software**

The SCSI directory in the C:\ drive contains the device drivers and troubleshooting utilities for the Adaptec SCSI controller (2.1.1(e)). The device drivers support a variety of hardware devices such as CD-ROM (2.1.1(f)), storage (2.1.1(d)), and tape drives (2.1.1(g)).

#### **(b) Backup Software**

The Backup software is used to save the user's data onto a 4mm tape (2.1.1(g)). Two types of backup software are available. The software located in the **NOVABK** directory allows the user to create a backup copy of the hard drive.

To execute the NOVABK software, type **novaback** at the DOS prompt.

The software located in the **NTAPE** directory is used to copy the user files in a UNIX tar tape format.

Note: The compression feature provided by the software or hardware must be disabled.

To execute the NTAPE software, type **ntape** at the DOS prompt.

(c) **Scanning Software**

The scanning software is a Microsoft Windows 3.1 application. It requires a device driver to be installed for the scanner.

- **PHOTOSHOP LE**

The software is used to scan images from hard copy sources such as photographs. It also provides the necessary tools to manipulate these images.

To execute the software, in MS Windows 3.1, double click on the Photoshop Limited Edition icon.

- **ENVISION Directory**

This directory contains the device driver for the scanner which should be installed in the system CONFIG.SYS file.

### **2.2.3 Directories for User Data**

The objective of these directories is to separate the user data from the application program files.

(a) **3DMODEL Directory**

This directory contains the three-dimensional library of models generated using the AutoCAD modeling software.

Note: The latest library of three-dimensional models is in the UNIX Workstation /TEC/models/ITD/DXF directory.

(b) **OLDMODEL Directory**

This directory is an obsolete model library. It should be removed if it still exists in the root (C:\) directory.

(c) **3DXF Directory**

This directory contains the DXF format files generated from the original AutoCAD DWG files.

Note: The latest library of three-dimensional models is in the UNIX workstation /TEC/models/ITD/DXF directory.

#### **2.2.4 Operating System**

The operating system software manages the hardware resources, and controls the operation of the application software. The PC modeling workstation contains two operating systems. Either of these can be selected when the system is being initialized.

##### **(a) Microsoft MS-DOS 6.0**

The DOS operating system is required by the software applications discussed thus far. The scanning software, Photoshop LE, also requires Microsoft Windows 3.1.

Note: The PC workstation is not equipped with Microsoft Windows 3.1.

##### **(b) Windows NT**

The Windows NT is a 32-bit high performance operating system. It ostensibly supports most applications written for Microsoft Windows 3.1. Only Microstation has been installed to take advantage of NT.

Note: Microsoft Windows 3.1 requires MS-DOS but Windows NT does not. In addition, the installation of device drivers in Windows NT and DOS follows different procedures.

### 3. UNIX-BASED MODELING STATION

#### 3.1 Hardware

##### 3.1.1 Computer Workstation

The Silicon Graphics Indigo 2 Extreme workstation provides the necessary computational power for developing and manipulating detailed three-dimensional wireframe models. It is also capable of displaying real-time renderings of textured models. However, its processing capability is not sufficient for the creation of highly populated built-up areas, or for real-time renderings of these models.

The SGI Indigo 2 Extreme system consists of:

- (a) MIPS R4400/150 MHz Processor
- (b) 128 Megabytes of Main Memory
- (c) 2.0 Gigabytes of System Disk
- (d) External r2 Square CD-ROM Reader
- (e) External r2 Square DAT Tape Drive
- (f) 3 1/2" Disk Drive
- (g) Extreme Graphics Card

#### 3.2 Software

##### 3.2.1 Visualization and Development Software

The purpose of the various software tools is to aid in the development, management and viewing of the model library.

###### (a) IRIS Performer Graphics Development Environment

IRIS Performer provides real-time rendering capabilities and graphic libraries (GL) programming. A modified version of the Performer demo program, **perfly**, is used to display static models in the model database.

To execute this program, type **perfly path/filename.extension** at the IRIS prompt.

###### (b) AutoCAD Release 12 for SGI

Like the DOS version, it is used for creating wireframe models of ITD or any other real world features.

To execute AutoCAD, type **acad** at the IRIS prompt.

Note: This software has upgraded to Release 13. To execute AutoCAD Release 13, type **acadr13** at the IRIS prompt.

(c) IRIS Development Option 5.2

The Development Option contains the C-compiler and additional graphics and system libraries to develop executable binary code for the SGI IRIX 5.2 UNIX based operating system.

(d) Informix SE, ESQL/C and SQL Software

The Informix SE is the Standard Database Engine; the ESQL/C is a set of compilers and libraries that translate and compile the embedded SQL/C programs; and the SQL is a database management system. These software packages provide the database and compiler interface functionality necessary to develop and operate the model library and its DBMS software.

To use the character based interactive mode of the Informix SQL DBMS system, type **isql** at the IRIS prompt.

Note: The Informix software installed are for single user only. If multiple users have to access the database simultaneously, multiple run time options need to be purchased.

(e) Showcase Application

The Showcase software tool comes with the operating system. It is a 3-D modeling software and is used to apply texture mapping and to translate DXF models to Inventor IV format files.

To execute Showcase, double-click the Showcase icon in the Demo-Applications window.

(f) Ez3d Modeler Software

Like Showcase, it can be used to map texture to the DXF models. Ez3d also provides tools needed to effectively and efficiently create 3-D wireframe representations of real-world objects. In addition, it can read and write in a variety of graphic file formats.

To execute Ez3d, type **Ez3d** at the IRIS prompt.

### 3.2.2 Data Directories

The purpose of these directories is to separate the application programs from the user generated files. This category includes the support directories needed by the textured model library.

The following are directories that constitute the model library:

- (a) /TEC/models/ITD/DXF contains AutoCAD DWG and DXF format files.
- (b) /TEC/models/ITD/IV contains Inventor format textured files.
- (c) /TEC/models/ITD/OBJ contains Wavefront format textured files.

The following are the directories containing the texture maps and the material library:



- (a) /TEC/models/ITD/textures contains the texture maps.
- (b) /TEC/models/ITD/material/ITD.mtl is the material library file for OBJ models.

### **3.2.3 MINFO DBMS Software**

MINFO is a database management system with a X-Window/Motif graphical user interface. The software was developed by MANDEX, Inc. to provide a set of tools for database manipulation and management. It has the facilities to search, edit and add information as well as for generating reports and viewing the library models in real-time. The user should refer to Section 5.3 for detail about this software.

To execute MINFO, type **minfo** at the IRIS prompt.

## 4. 3D TERRAIN FEATURE MODELS

### 4.1 Model Category

The three-dimensional models created under this Phase II SBIR project are representations of the terrain features defined by the DMA's ITD set. These features are classified into the following six sub-categories by the ITD:

- Obstacles (OBS)
- Slope/Surface Configuration (SLP)
- Soil/Surface Materials (SMC)
- Surface Drainage (SDR)
- Transportation (TRN)
- Vegetation (VEG)

In the model database, each feature model is assigned to a category (ITD) and a sub-category (one of the six above). The models are named using their Feature Attribute Category Code (FACC) followed by a version number. For example, **AL260** is the FACC code for Wall under the obstacle sub-category, while **al260\_01.dxf** is the DXF model file name for a cross buck wooden fence which falls into the description for the Wall.

*Appendix A* lists all models that are currently in the model database according to their ITD sub-categories. Some models are available in different file formats (DXF, IV, OBJ) as indicated in the appendix.

### 4.2 Texture Maps

Texture mapping is to map an image, either digitized or synthesized, onto a surface to add surface detail. The technique was pioneered by Catmull<sup>[3]</sup> and refined by Blinn and Newell<sup>[4]</sup>. Many algorithms have been developed to accomplish texture mapping. Heckbert<sup>[5]</sup> provides a thorough survey of texture-mapping methods.

Texture mapping technique is often used to increase the realism of a 3D computer model. However, performing texturing at interactive rates places high demand on the display system and is feasible only in high-performance workstations. To meet the various needs of the modeling and simulation community, we created two sets of ITD feature models, the non-textured and the textured models. Users with relatively low-performance hardware systems should use the non-textured models in their simulation which would provide a reasonable rendering speed, while users with powerful workstations may choose the textured models to generate more realistic scenes.

To locate useful texture maps, MANDEX performed a search over the Internet. Among many sources, the largest collection found is free of charge from MIT's Vision Texture 1.0 database. The database consists of more than 100 texture maps which can be divided into the following categories:

Bark	Brick	Buildings	Clouds	Fabric	Flowers
Food	Grass	Leaves	Metal	Misc	Paintings
Sand	Stone	Terrain	Tile	Water	Wood

Two standard image sizes are available: 128x128 and 512x512. We downloaded all the 128x128 images. These images are in **ppm** format. To use these images in IRIS Showcase or Ez3d software, they were converted to **rgb** format using IRIS Imagetool **fromppm**. These images can be found under the directory: **/TEC/texture/VisTex/FLAT/128x128**.

Other available texture maps that come with the SGI system are located in the following directories: **/usr/share/data/textures** and **/usr/demo/data/i3dm/textures**. These images were used to apply texture to the terrain feature models.

### 4.3 Texture Mapping and Format Conversion

Most of the terrain feature models were created using AutoCAD software and saved in its native format (DWG/DXF). Since these formats do not support texture, the textures were added to the models using other software. When selecting software for texture mapping, the following factors were considered:

- (1) The software must have texture mapping capability.
- (2) The software must recognize DXF format.
- (3) Which output file formats are supported by the software.

Items (1) and (2) are required in the case of adding textures to DXF models. From our experience, most 3D modeling software has the capability of importing DXF models. The difficult part is to find a software application that can export the resulting textured model to the necessary or required format, such as OBJ.

Two different software packages were used for texture mapping. The first is the IRIS Showcase. A DXF model can be loaded into the software, edited (add texture to it), and saved in the SGI Open Inventor format (IV). The IV format is the only output format this software supports. The feature models in IV formats were generated using the Showcase software. The second software package we used was the Ez3d Modeler by Radiance Software International. It claims to support various input and output formats; however, we tested only the DXF, IV, and OBJ importer and exporter. The OBJ models in the model database were generated using the Ez3d software along with some manual editing to fix the bugs in the OBJ export translator. Detailed procedures for adding textures using this software are given in the following sections. The emphasis is on the problems encountered during the process and the actions taken in solving these problems.

#### 4.3.1 IRIS Showcase

This software comes bundled with the SGI Indigo 2 workstation system software. It is one of the tools used to convert the original DXF models to Open Inventor IV format. The conversion process is as follows:

### Procedures:

- (1) In the Demo Applications Catalog, double click the Showcase icon, or at the IRIS prompt, type: **showcase**
- (2) Under the **File** menu, select **Insert**, and then choose **3D Models**.
- (3) Select file to insert (e.g. /TEC/models/ITD/DXF/ec030\_01.dxf) and click on **Accept**.
- (4) A window appears, place it by clicking the left mouse button.
- (5) To apply texture to the DXF model, double click on the model.
- (6) Under **Gizmos** menu, choose **3D** and then select **3D**.
- (7) Again under **Gizmos**, choose **3D** and then select **Texture**.
- (8) In the 3D Gizmo window, turn texture on and choose a texture set.
- (9) Select the DXF model by clicking on it.

Note: Sometimes, it may be necessary to ungroup the DXF model into individual polygon faces before applying texture.

- (10) From the texture set, choose a texture pattern.

Note: Immediate results can be observed in the Texture Gizmo window. The appearance of the mapped image can be modified by changing the default settings in this window.

- (11) Under **Edit** menu, select **Done Editing**.
- (12) From the **File** menu, choose **Save As 3D (Inventor)**.
- (13) Name the file with an .iv extension (e.g. /TEC/models/ITD/IV/ec030\_01.iv) and click on **Accept**.

### Problems and Possible Solutions:

- (1) When ungrouping DXF models, showcase separate entities by color. In other words, all polygon faces of the same color and with common edges remain together.

To overcome this limitation:

- Use AutoCAD to change the color of the polygons sharing edges and color.
- Then, create a new DXF file and bring it into Showcase.
- After ungrouping the model, apply texture to individual faces.
- Finally, assign the same original color (material) to the polygon faces.

- (2) There are cases when applying texture creates texture maps that look as if they were stretched. Following the previous steps, (1) sometimes helps correct this problem.
- (3) In addition, polygons with unusual orientations in 3-D space create distorted texture maps. At the time of this writing, no solution for this type of problem has been found.

#### **4.3.2 Ez3d 1.0**

The Ez3d Modeler software was purchased by TEC to be used as a tool to add textures and convert 3-D models from one format to another. This software was used to put textures onto the DXF models and save the textured models in the Wavefront OBJ format. The software's native format is an extended form of the standard SGI Inventor 1.0 format. Models in the following formats can be imported into the software:

- Alias
- AutoDesk DXF
- Inventor
- Open Inventor
- 3D Studio
- IGES
- SoftImage
- Wavefront OBJ

Only the DXF and the Inventor/Open Inventor import translators came bundled with the Ez3d software. Import translators for other formats can be obtained from the Abaco System's FTP site:

**via.net  
/pub/acuris/ivbeta1**

These translators were developed by Abaco System, Inc. (formerly Acuris Software), and licensed to SGI for distribution to SGI customers.

We downloaded only the OBJ translator from the above site. It was installed in the appropriate directory according to the instructions given by the Ez3d on-line help file. Among the above import translators, we used only the DXF, Open Inventor, and the OBJ translator.

The models created or edited using Ez3d can be saved in any of the following formats:

- DXF
- Inventor
- Open Inventor
- Wavefront OBJ
- RenderMan

Except for the RenderMan translator, all other export translators are included in the software. Again, we tested only the DXF, Open Inventor, and Wavefront OBJ translator.

The following is a step-by-step procedure for using Ez3d to add textures to a DXF model and save it in OBJ format.

## Procedures:

- (1) Run Ez3d from command line by typing:  
**rehash** (only the first time)  
**Ez3d**
- (2) Open or select a Ez3d Project (see Ez3d Manual<sup>[6]</sup>).
- (3) Choose the **Sculptor - Load** option from the main menu to load the DXF model.
- (4) Select the model parts on which you want to put texture.
- (5) Choose the **Costume - Texture** option from the main menu to select and apply texture to the selected model parts.
- (6) Select the entire model using the **Edit - Select All** option from the main menu.
- (7) Choose the **Sculptor - Save** option and select Wavefront OBJ as the file format to save the textured model in OBJ format.

The saved filename will have the form "Name.Type.Format." The user need only type in a name (Example: "car") and Ez3d will automatically add the extensions for file type and file format. In our case, the saved filename will be: "car.object.obj." There are three different types of files in Ez3d: object, scene, and spline files. We are only interested in the object files. For OBJ models, two other files are generated by Ez3d: the material library file "car.object.mtl" and a symbolic link to the texture map used by the model. More symbolic links will be generated if more than one texture map is applied to a model.

## Problems and Possible Solutions:

- (1) When the resulting OBJ model was viewed using IRIS Performer, the texture did not show up. We traced this problem to the material file, where the corresponding texture map was never referenced. According to Radiance Software International, Inc., this is a bug in the Ez3d 1.0 software, and it will be fixed in Version 2.0, which will be available soon.

To solve this problem, you need to edit the corresponding material file (.mtl file) to add the reference to the texture map. This can be done by inserting a line under the material definition for the textured model parts. For example, for the "car" model mentioned previously, if a texture map (bumper.rgb) has been put on its bumper, and the material used for the bumper is called "Material\_1" in the "car.object.obj" file, you must edit the material file "car.object.mtl," and insert a line below the definition for "Material\_1":

```
newmtl Material_1
map_Kd Bumper.rgb          (inserted line)
Ka 0.095314 0.095123 0.095532
Kd 0.372322 0.371574 0.373173
Ks 0.890909 0.887832 0.890909
```

illum 1  
Ns 127

- (2) The saved OBJ model was in a different orientation than the original DXF model. This was a result of using different coordinate systems. Ez3d uses y as the vertical axis, while the models use z as the vertical axis.

While inside Ez3d, rotate the entire model 90 degrees around the x axis before saving. By doing so, the resulting OBJ model will have the same orientation as the original DXF model.

- (3) When a DXF model was imported into Ez3d, it could not be broken down to individual faces (polygons). This created some problems for attaching textures. IRIS Showcase had the same problem. We suspect the problem is in the DXF import translator. It groups the faces according to certain properties, and we were unable to determine what rules govern this grouping.

We do not have a good solution to this problem. It seems that if we use different colors for adjacent faces in the DXF model, these faces can be separated in Ez3d. This is also true for Showcase.

- (4) When a model was partially textured, the numbering for the texture vertices in the resulting OBJ file was sometimes wrong. This was caused by not numbering the three types of vertices (geometry, normal, and texture) independently, as defined by the OBJ format. It is a bug in the OBJ export translator.

To get around this problem, you must group together the parts with textures **first**, then group the non-textured parts before saving the model. This step ensures the correct numbering for texture vertices.

- (5) When viewed using Performer, some OBJ models had "missing" faces. Those faces did not show up because the order of the vertices as well as the directions of the corresponding normal vectors were wrong. Since this problem does not happen for every model, we don't know what caused it.

When this problem happens, we had to go into the .obj file, locate the missing face, change the sign of the corresponding normal vectors, and reverse the order of the vertices that defined the face. This is a very tedious process to go through. We do not recommend doing so in the future.

Appendix C shows an example of using Ez3d to add texture, and to convert a DXF model into OBJ format. Users should refer to the appendix for detailed procedures. In addition to the problems listed above, the OBJ import translator translates only the model's geometry, not the color (material) and texture. When loading an OBJ model to Ez3d, the model color is defaulted to blue. Since we are not converting OBJ models to other formats using Ez3d, this problem does not affect us in any way. It is mentioned here for future reference.

In summary, when used as format converter, Ez3d Modeler 1.0 has many problems, most of which reside in the import/export translators. Unless they are fixed in the next release of the software, it is not a good tool to use. One way to minimize problems is to create textured models using Ez3d, and then save them to the desired formats. This way, the problems associated with the import translators will no longer exist, however the problems caused by the export translators will still exist.

The two software packages discussed previously are modeling packages similar to AutoCAD, but support additional features, such as materials and textures. We used only a small portion of the software to put textures on our DXF models and converted them to the desired formats. Other functions of the software are not relevant to our task and are therefore not mentioned. Users should refer to the appropriate documentation for information about the usage of each software package.



## 5. 3D MODEL DATABASE

Building a standard library of 3D simulator models is one of the main goals of this SBIR Phase II project. The database and DBMS software design objectives are: (1) to provide TEC with the capability of storing, categorizing and organizing 3D models; and, (2) to develop a set of tools to assist in updating and maintaining the database, as well as distributing models to customers.

### 5.1 Database Structure

The model database is a relational database implemented using the Informix Standard Engine (Informix-SE 5.03) software from Informix Software, Inc. The database is called **models**. It consists of the following six tables:

<b>model_info</b>	Main table that stores model information
<b>country</b>	Country names and codes
<b>service</b>	Service names
<b>branch</b>	Branch names
<b>category</b>	Model major category
<b>distribution</b>	Model distribution codes and descriptions

The data elements contained in each of the six tables are listed in *Appendix B*. The database is physically located in **/TEC/database**. The owner of the database and tables is the database superuser **informix**. The SQL scripts used to create these tables are located in **/TEC/database/sql** and have the file format <table name>.sql. For example the SQL script used to create the **model\_info** table is called **model\_info.sql**.

### 5.2 Database Backup and Recovery Procedure

In addition to the usual system backup, the following procedure is a simple yet useful way of restoring data in the case of a database crash:

- (1) Login as **informix**;
- (2) Change directory to **/TEC/database/sql**; then do either step (3) or (4) below;
- (3) To backup the database, type the following command:

**isql < unloaddb.sql**

As the result of this command, the data in each table are saved in a file named <table name>.dat, and can be reloaded into the database using step (4) when necessary.

(4) To restore data saved using step (3), type the following command:

```
isql < loadddb.sql
```

Step (4) should be performed with caution because it will empty each table before loading the backup data. The user should be sure that step (3) was executed successfully the last time it was performed. Any changes made after the last backup will be lost as a result of step (4). We suggest that user perform steps (1) through (3) as frequently as deemed necessary to keep an up-to-date backup data set. Perform step (4) only when it is absolutely necessary.

### 5.3 DBMS Software

The database management system software, MINFO, provides tools for the user to interact with the model database, such as add/delete models, look-up and view models, update model information, generate reports, etc. The software is menu driven with a user friendly graphical user interface.

#### 5.3.1 Preparing to Use the Software

The source code and executables for the DBMS software are located in `/TEC/database/src` directory. The directory contains the following files:

<code>Makefile</code>	-- Main make file
<code>model_info.ec</code>	-- Main program written in ESQL/C
<code>db_access.ec</code>	-- Functions written in ESQL/C
<code>model_info.uil</code>	-- Graphical user interface definition
<code>model_info.h</code>	-- Header file for the ESQL/C programs
<code>struct_def.h</code>	-- Header file for the ESQL/C programs
<code>identifiers.uih</code>	-- Header file for the user interface program
<code>procedures.uih</code>	-- Header file for the user interface program
<code>values.uih</code>	-- Header file for the user interface program
<code>minfo</code>	-- DBMS software executable file
<code>perfly</code>	-- Symbolic link to the Performer viewer executable file
<code>minfo.ace</code>	-- SQL ACE report

Use the **Makefile** to compile the programs and generate the executables after each modification to the source code.

Before you can use the software, you must be sure that your user environment is set up as required by INFORMIX and the DBMS software. This can be done through the following steps:

(1) Edit your **.login** file to add the following lines:

```
setenv TERM xterm
setenv TERMINFO /usr/lib/terminfo
setenv INFORMIXDIR /usr/informix
setenv PATH ${PATH}:${INFORMIXDIR}/bin
```

```
setenv SQLEXEC $INFORMIXDIR/lib/sqlexec
setenv INFORMIXTERM terminfo
setenv DBPATH /TEC/database
```

- (2) Copy the **.Xresources** file from **/usr/guest** directory to your home directory.
- (3) Ask the Database Administrator (DBA) to grant you the permission to access the **models** database. This step is not necessary in the current environment, where the designated DBA for the **models** database is user **vgupta** (Vineet Gupta), and every user on the system (the public) has a connection permission to the database.

Once you have completed the above steps, you can run the software from any working directory by enter the following command from the command line:

**minfo**

### 5.3.2 User Guide

This section servers as a user guide to the DBMS software. As mentioned earlier, the software is menu driven. The menu bar is located at the top of the GUI window. It contains ten options: Query, Add, Update, Delete, Next, Prev, List, Copy, Accessory, and Miscellaneous. These options allow users to perform basic database operations easily and efficiently. The function of each menu option is described in detail bellow.

#### (1) Query

Querying means searching for information that is stored in the database. Once the **Query** option is selected, the data entry area on the screen will be enabled to allow you to enter the search criteria. You may base your search on any data elements stored in the **model\_info** table. To enter the search criteria, move the cursor to the corresponding data entry field and type the data you want to match. The completed Query Screen tells the program to search for rows in the database in which the corresponding columns match the data you have entered. Leaving a field blank tells the program not to consider it as part of the query.

To initiate the search, press the **OK** button on the control panel located on the right side of the screen. To cancel a search, press the **Cancel** button.

For character data field, you may also use a wildcard as part of your search criteria. The wildcard character is the asterisk (\*). You can use the \* to match any group of zero or more characters. For example, to find models whose names include the word "TREE", enter \*TREE\* in the model name field.

The result of a query may contain zero or more matches. If no rows were found that match the search criteria, a message box will pop up to inform you. Otherwise, the information about the first model that matches the search criteria will be retrieved from the database and displayed on the screen. If more than one model matches, the list is sorted in the order of country, service,

branch, category, sub-category, model ID, and version. You can use the **Next** and **Prev** menu options to browse through the list.

Once a model is displayed on the screen, the **View** button on the control panel will be enabled. Activation of this button triggers the 3-D visualization software (Performer or Inventor) which displays the corresponding model in its three-dimensional form. For models in Open Inventor format (.iv), the inventor viewer **ivview** will be invoked. For other formats, the Performer viewer **perfly** will be invoked. The reason for using **ivview** to display an inventor model is that the Perform IV loader does not translate the texture. Data formats supported by IRIS Performer are listed in **Table 5-2**.

**Table 5-2.** IRIS Performer Supported Data Formats

Name	Description
BIN	Silicon Graphics data format
DWB	Designer's Workbench format
DXF	AutoCAD® format
FLT	MultiGen FLIGHT format
GFO	Silicon Graphics data format
IV	IRIS Inventor format (ASCII format)
LSA, LSB	Formats from Lightscape Technologies, Inc.
OBJ	Wavefront Technology <i>Model</i> format
POLY	Polyhedron data format
PTU	IRIS Performer terrain utility format
SGO	Silicon Graphics graphic object data format
STL	3-D-lithography interchange format
SV	I3DM SuperView data format

It should be pointed out that the Performer file loader may not translate all of the features a data format has. We tested only the DXF, IV and OBJ loaders. Many features supported by the DXF format are not translated. The IV loader does not translate texture. By comparison, the OBJ loader seems to be the best among the three.

## (2) Add

Use the **Add** option to create a new record in the **model\_info** table. Upon selection, the screen cursor is moved into the first data entry field. Enter the data you want to add to each data field. Press the **TAB** or **ENTER** or ↓ key to move the cursor to the next field. The **SHIFT+TAB** or ↑ key takes you back to the previous field in case you need to go back and reenter the data.

Some fields require data entry. These fields are:

Country  
Service  
Branch  
Category  
Model Name  
Path and File Name  
Distribution

The program will not allow you to save the data if any of these fields are left blank. Validation checks are performed on some fields to be sure the data entered are valid. There are look up lists for Country, Service, Branch, Category and Distribution fields. Use the **F2** key to display the look up list while the cursor is in one of these field. Only the items shown in the list are considered valid entries. If you could not find the appropriate item from the displayed list, a new record must be added to the list. Use the **Miscellaneous** menu to add new Country, Service, Branch and Category records to the corresponding database tables.

Press the **OK** button to insert the new record into the database after you complete the data entry, or press the **Cancel** button to abort.

### **(3) Update**

The **Update** option works the same way as the **Add** option except that it is used to modify an existing record in the database. This option is not enabled until you select a record using the **Query** option. Press **OK** to save the changes. **Cancel** will take you out of the data entry mode without modifying the record.

### **(4) Delete**

This option allows you to delete a record from the **model\_info** table. To be able to access this option, a model has to be selected first. Only the record currently displayed on the screen will be removed from the database. A message will appear on the screen to ask you to confirm the deletion before any action is taken. This will prevent you from accidentally deleting a record.

### **(5) Next**

The **Next** option displays the next record in your selection list, if there are more than one. Along with the **Prev** option, it provides a way for you to browse through a list of records that satisfied your search criteria.

### **(6) Prev**

This option allows you to go back to see a previous record on your selection list.

### **(7) List**

The **List** option writes the model file name (with full path) and its dimensions to a user specified file. The user has the option to save either the current displayed model only, or all the models on

the selection list. This option can be used as an interface between the model database and DrawLand, a software package being developed by TEC. DrawLand can call up this software to select models. The information saved in the given file can then be read by DrawLand and passed to the model placement tools.

### **(8) Copy**

This option was designed to simplify the process of distributing models to customers. The **Copy** option makes a new sub-directory under your current working directory and copies all selected models including related material and texture files to this new sub-directory. The sub-category of the first model on the selection list will be used as the sub-directory name, therefore, it makes more sense if you select models according to their sub-categories, and do this one sub-category at a time. Once completed, the files in this new directory can be transferred to customer site or copied to a tape. Be sure to delete the entire directory once the distribution is made.

The program assumes that the material and texture files for a given model category are located in the directories **/TEC/models/<category>/material** and **/TEC/models/<category>/texture**, respectively. For example, the material and texture files for the ITD models are in **/TEC/models/ITD/material** and **/TEC/models/ITD/texture**. Similar directory structure should be used in the future for other models in order for the **Copy** option to find those files.

### **(9) Accessory**

The **Accessory** option checks to see if there are any accessories associated with the currently displayed model. If there are, a list of accessories will be displayed on the screen, otherwise a message will appear telling you no accessory was found for the current model. An accessory is also a 3-D model itself and has a corresponding record in the **model\_info** table. The relationship between the main model and its accessory is that they belong to the same sub-category, but the accessory has a "Y" in its accessory indicator field, while the main model does not.

### **(10) Miscellaneous**

The **Miscellaneous** option provides additional tools to manipulate and manage the database. It has a pull down menu that contains four items:

- Add Country**
- Add Service**
- Add Branch**
- Report**

The first three items allow you to insert new records into the **country**, **service**, and **branch** table in the **models** database. The country, service, and branch associated with a model must be defined in these tables, otherwise they are considered as invalid data by the program.

The **Report** option generates a hard copy of all information for selected models in the form of a report. The report is saved as an ASCII file **minfo.rpt** in the user's working directory. It can be directed to any printer using the appropriate commands.

## 5.4 Database Management Using INFORMIX-SQL

INFORMIX-SQL is a Relational Database Management System by Informix Software, Inc. It consists of useful programs or modules that perform data management tasks. The database administrator can use this tool to maintain the INFORMIX database. To start the system, enter the following command at the system prompt:

```
isql
```

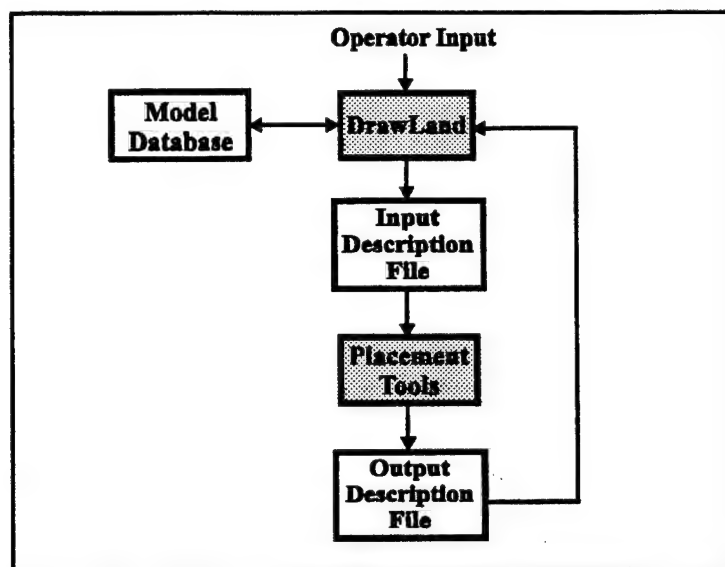
This operation displays the INFORMIX-SQL Main Menu. The options on the Main Menu give you access to all INFORMIX-SQL functions. Select an option on the Main Menu, and INFORMIX-SQL displays a new menu of options relating to your choice. For example, if you select the **Table** option from the Main Menu, you see another menu that lets you specify whether you want to create, alter, display information about, or drop a database table. Here is a summary of the INFORMIX-SQL Main Menu options.

<b>Form</b>	Displays the FORM Menu. Use this menu to work with screen forms.
<b>Report</b>	Displays the REPORT Menu. Use this menu to work with ACE reports. A report selects data from one or more tables in a database and prints it in the format you specify.
<b>Query-language</b>	Displays the SQL Menu. Use this menu to work with the SQL query language. You use this language to search a database for information, add information to a database, and change the structure of a database.
<b>User-menu</b>	Display the USER-MENU Menu. Use this menu to work with custom screen menus.
<b>Database</b>	Displays the DATABASE Menu. Use this menu to select, create, or drop a database.
<b>Table</b>	Displays the TABLE Menu. Use this menu to create, alter, or drop a table.
<b>Exit</b>	Exits INFORMIX-SQL and returns to the operating system.

Since INFORMIX-SQL is a commercial product, its usage will not be discussed in detail in this report. The user should refer to the product user guide and reference manuals for how to use the software.<sup>[7, 8]</sup>

## 6. DYNAMIC MODEL PLACEMENT TOOLS

The dynamic model placement tools were developed to enhance DrawLand's capability of manipulating and placing models into the simulation scene. The software was designed as an independent executable program which interfaces with DrawLand or other software through input/output description files. The interface of the placement tool with DrawLand is shown in *Figure 6-1*. The program can be modified to become functions of DrawLand.



*Figure 6-1.* Interface between DrawLand and the model placement tools.

The DrawLand operator will select a model from the model database, specify the desired type of placement (Point, Linear or Area), and give the desired model locations in the scene. These operator input parameters will be written to an input description file. The placement tool reads the parameters from the input description file, and calculates the total number of copies of the given model needed to be placed, and their locations, orientations and scaling factors, using the appropriate algorithm for the specified type of placement. The results are written to the output description file. DrawLand software reads this file and displays each copy of the model according to its given location, orientation and scaling factor.

For all three types of placement, the operator specifies the desired model locations in terms of a series of two dimensional points (x-y coordinates). For point placement, these points represent the exact locations at which models should be placed. For linear placement, they define a path along which models should be placed according to the specified inter-distance. For area placement, they define the boundary of an area within which models should be placed according to the specified density. Algorithms were developed to accomplish the above mentioned three types of placement.



## 6.1 Input Specification File

The input description file contains the parameters specified by the DrawLand operator. It must be in the format described in *Table 6-1*.

*Table 6-1*. Format of the input description file.

! format.in	
! Comment starts with a !	
! There can be as many comment lines as you like in the beginning of the file but not between	
! data lines	
!	
2 point	! number of operator specified points:
p	!feature type: p=point, l=linear, a=area;
/TEC/models/ITD/OBJ/bougus.obj	!model to place copies of:
2.25 0.125 1.5	!model dimension in x y z (in meters):
1.00 1.00 1.00	!x y z scaling factor;
0.1	!size variation tolerance (for area placement);
2.25 c	!model spacing: c=constant, r=random;
18 0 0 E	!origin Longitude;
43 0 0 N	!origin Latitude;
1030 4467	!offset of point 1 from origin (in meters);
1020 4367	!offset of point 2 from origin (in meters);

Although the same file format is used for all types of placement, some parameters are only used by certain types of placement. For example, the model size variation tolerance and the constant/random spacing indicator apply only to the area feature placement. They are ignored by point and linear feature placement algorithms.

## 6.2 Output Specification File

The output description file is created by the model placement program. It contains information such as the number of copies to be placed, the location, orientation and scaling factors for each copy, as well as others given by the input description file. The format of the output description file is given in *Table 6-2*.

In the output description file, the model altitude (z) is the origin of the model relative to the surface of the ground. The yaw, pitch and roll determine the orientation of the corresponding model. For point and linear placement, the x y z scaling factors are the same as those given in the input description file. But for area features, these factors are varied randomly by an amount up to  $\pm$ (size variation tolerance) to produce a random look in model sizes.

**Table 6-2.** Format of the output description file

! format.out	
! Comment starts with a !	
! There are no comments in between the data lines	
!	
2 models	! number of copies to be placed:
18 0 0 E	!origin Longitude;
43 0 0 N	!origin Latitude;
/TEC/models/TTD/OBJ/bougus.obj '	!model to place copies of;
1030 4467	!offset of copy 1 from origin (in meters), x y;
0	!height from ground (in meters), z;
0 0 20	!yaw, pitch, roll (in degrees);
1.0 1.0 1.0	! x y z scaling factors;
1020 4367	!offset of copy 2 from origin (in meters), x y;
0	!height from ground (in meters), z;
0 0 30	!yaw, pitch, roll;
1.0 1.0 1.0	! x y z scaling factors;

## 6.3 Model Placement Algorithms

### 6.3.1 Point feature

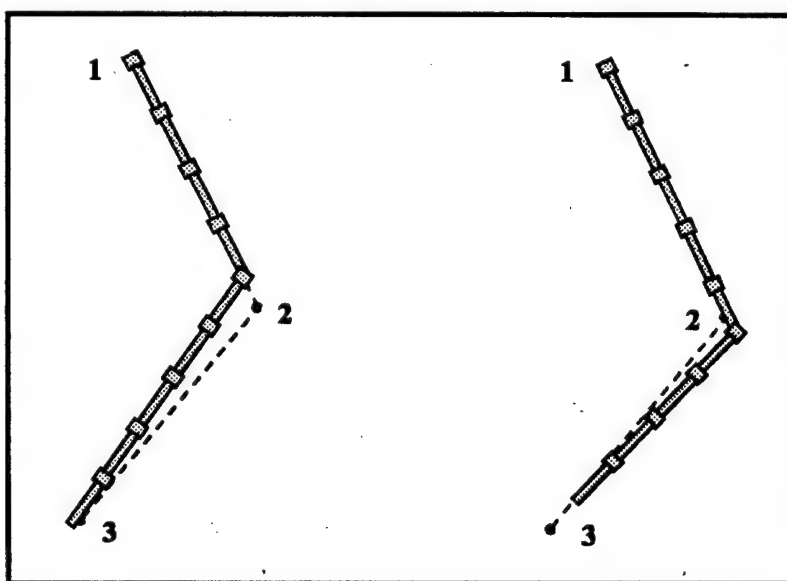
The point feature placement is a straightforward task. The program simply takes the operator specified points and uses them as the model origins. The number of copies to be placed is equal to the number of points given in the input description file. All other parameters are passed to the output description file without change. The yaw, pitch and roll for each copy of the model are defaulted to zero so that all copies have the same size and orientation. The only difference is in the model's location. The operator should make sure that the distance between two consecutive points specified is large enough (or use the proper scaling factor) so that the models will not overlap with one another. The data for size variation tolerance and model spacing are not used by the point feature placement algorithm.

### 6.3.2 Linear feature

Many terrain features can be constructed by following a given curve and concatenating basic model segments, such as the construction of a fence line, a road, etc. The linear feature placement tool was developed to automate this process. The user specified points are connected by straight lines to form a "path". Models will be placed along this path at a specified inter-distance.

The algorithm assumes that the model segment is oriented in such a way that it is extensible in the x-direction (longitude direction). The tool may be used for two purposes: (1) to construct linear features, such as fence lines and roads, and (2) to place a series of point features, such as lines of trees. In the first case, the spacing between models is crucial. It must be equal to the model dimension in x-direction in order to avoid gap or overlap between models. While for the second case, the spacing only determines how far apart the models will be placed. As long as the spacing is not too small, the outcome of the placement will look reasonable.

The actual distance between models is determined by the model spacing multiplied by the x scaling factor given in the input description file. For each line segment in the path, the algorithm computes the number of models that can be fit in and calculates the yaw angle for these models to line up in the correct orientation. The algorithm handles the "corner" in the following ways: if the remaining portion of the line segment is greater than one half of the model length, the line segment is extended to fit an extra model. The next line segment will start from the end point of this additional model. Otherwise, the remaining line segment is discarded and the next segment starts at the end of the last model. In doing so, the algorithm does not leave a gap at the corner, and all model segments have the same size. The trade off is that the corner points are deviated from the operator specified locations by an amount up to one half of the model length. The two scenarios that might occur at the corner of two line segments are illustrated in *Figure 6-2*.



**Figure 6-2.** Two scenarios that might occur at the corner of two line segments.

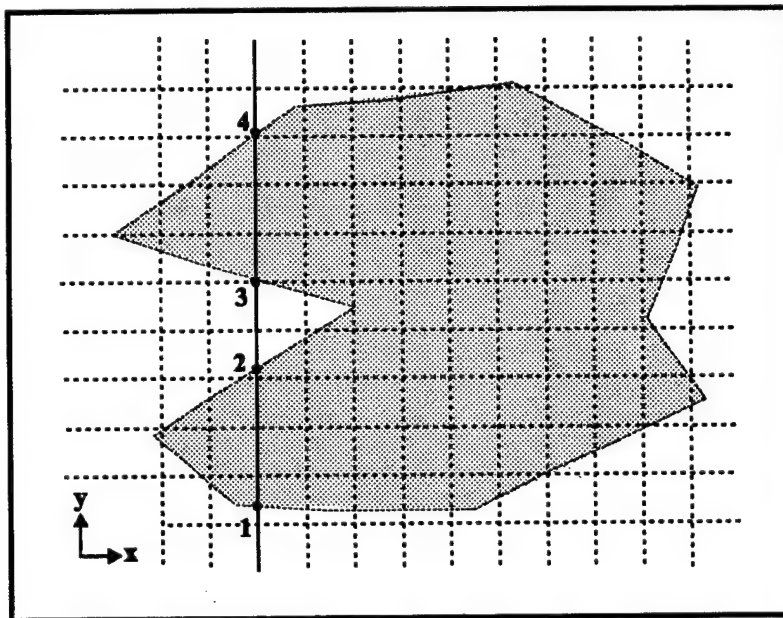
It is important to point out that this generic algorithm works better for certain types of features than others. For example, the algorithm produces a perfect fence line but a not so perfect road. At the turn (corner of the path), the road segments are jointed at the center line, leaving a gap on one side and an overlap on the other. One solution to this problem is to add a single model segment with the appropriate turning angle at each location where the road turns.

Another limitation of this algorithm comes from the fact that the path specified by the DrawLand operator is only two-dimensional. The lack of elevation data (z coordinates) forces the algorithm

to make the assumption that the features are on a horizontal plane. As a result, the model's pitch and roll angles are not calculated. When the models are lowered to the surface of the terrain, the joint between models may not look as good as when they were on a plane, depending on how large the variations are in elevation. The algorithm can be improved if a three-dimensional path is available.

### 6.3.3 Area feature

A typical area feature is a forest with trees of different sizes covering an area. The area feature placement tool was designed to automate the creation of such features. The operator specifies the boundary of an area through a series of two-dimensional points. The area can be of arbitrary shape. The algorithm first determines the bounding box of the given area: (xmin, ymin) (xmax, ymax), then it generates a regular grid over the bounding box with the grid size equal to the model spacing given in the input description file. For each x-grid value, the algorithm finds the corresponding boundary crossing points (*Figure 6-3*) and sorts them according to their y values. Models are then placed between every other consecutive boundary crossing point. For the example shown in *Figure 6-3*, models are placed between point 1 and 2, 3 and 4.



*Figure 6-3.* Boundary crossing points 1, 2, 3 and 4.

The current algorithm assumes that copies of a single model are to be placed. The variation in model sizes is achieved through the size variation tolerance given in the input description file. A 0 tolerance will result in models of identical size. A 0.1 tolerance will produce a random size variation between 90% and 110% of the actual model size. This parameter gives users control over the model size variations.

Besides the random size variation, the algorithm also allows users to choose between constant and random model spacing by setting the spacing indicator to c (constant) or r (random). The model locations are first calculated using a regular grid (constant spacing). If random spacing is desired,

a random number ranging from -40% to +40% of the given model spacing is added to the x and y coordinates of the model origin to produce a random look in model locations.

#### 6.4 How to Run the Tool

The model placement tools were programmed using C language and located in the directory **/TEC/tools**. The name of the source code is **ptool.c**. Use the **Makefile** located in the same directory to compile the file and generate the executable **ptool**. This executable file has been symbolically linked to a file by the same name in the **/usr/bin** directory. Any user who has **/usr/bin** included in their **PATH** environment variable should be able to execute **ptool** from any sub-directory on the system. To run the tool, type the following command at the UNIX command line (or call it from within another software):

**ptool** *in\_file out\_file*

In the above command, *in\_file* is the name of the input description file, and *out\_file* is the name of the output description file. The file names may include a full path. The user should make sure that he/she has the appropriate read/write permissions to the directories and the specified files.

## 7. REFERENCES

- [1] James, Gregory D. and Javier F. Maida, "Three Dimensional Modeling Station," Final Report, MANDEX, Inc., MDX-92-S-3409, 1992.
- [2] Fornshell, Ben, "A User's Guide to DrawLand," U.S. Army Topographic Engineering Center, 1995.
- [3] Catmull, E., "A Subdivision Algorithm for Computer Display of Curved Surfaces," Ph.D. Thesis, Report UTEC-CSc-74-133, Computer Science Department, University of Utah, Salt Lake City, UT, December 1974.
- [4] Blinn, J. F. and M. E. Newell, "Texture and Reflection in Computer Generated Images," *CACM*, 19 (10), October 1976, pp. 542-547
- [5] Heckbert, P. S., "Survey of Texture Mapping," *CG & A*, 6 (11), November 1986, pp. 56-67.
- [6] "Ez3d<sup>TM</sup> Modeler Version 1.0 User's Manual," Radiance Software International, Berkeley, CA, November 1994.
- [7] "INFORMIX-SQL User Guide," Informix Software, Inc., Menlo Park, CA, March 1990.
- [8] "INFORMIX-SQL Reference Manual," Informix Software, Inc., Menlo Park, CA, March 1990.

## APPENDIX A

### ITD FEATURE MODEL LISTING

#### A-1. Obstacles

ITD Feature Category	Model Filename	Model Description	File Format
<b>Dragon Teeth</b>	al060_01	Welded I-beam	DXF, OBJ, IV
	al060_02	Column	DXF, OBJ, IV
	al060_03	Truncated Pyramid	DXF, OBJ, IV
	al060_04	Pyramid with Detailed Chain	DXF, OBJ, IV
	al060_05	Jersey Wall	DXF, OBJ, IV
	al060_06	Pyramid with Simple Chain	DXF, OBJ, IV
	al060_07	Guard Rail with Left End Cap	DXF, OBJ, IV
	al060_08	Offset Guard Rail with Right End Cap	DXF, OBJ, IV
	al060_09	Offset Guard Rail with no End Cap	DXF, OBJ, IV
<b>Wall</b>	al260_01	Cross Buck Fence	DXF, OBJ, IV
	al260_02	Horizontal Slat Fence	DXF, OBJ, IV
	al260_03	Stacked Log Fence	DXF, OBJ, IV
	al260_04	Wire Fence on Wood Posts	DXF, OBJ, IV
	al260_05	Chain Linked Fence on Wood Posts	DXF, OBJ, IV
	al260_06	Picket Fence	DXF, OBJ, IV
	al260_07	Chain Linked Fence on Metal Posts	DXF, OBJ, IV
	al260_08	Block Wall	DXF, OBJ, IV
	al260_09	Brick Wall	DXF, OBJ, IV
	al260_10	3' Chain Linked Fence on Metal Posts with Barbed Wire	DXF, OBJ, IV

	al260_11	Concertina Wire	DXF, OBJ, IV
	al260_12	3' Chain Linked Fence on Metal Posts with Concertina Wire	DXF, OBJ, IV
	al260_13	8' Chain Linked Fence on Metal Posts with Concertina Wire	DXF, OBJ, IV
	al260_14	10' Chain Linked Fence on Metal Posts with Concertina Wire	DXF, OBJ, IV
	al260_15	8' Chain Linked Fence on Metal Posts with Barbed Wire	DXF, OBJ, IV
	al260_16	10' Chain Linked Fence on Metal Posts with Barbed Wire	DXF, OBJ, IV
	al260_17	Barbed Wire	DXF, OBJ, IV
	al260_18	Wood Retaining Wall with I-Beam Posts	DXF, OBJ, IV
	al260_19	Corrugated Steel Retaining Wall	DXF, OBJ, IV
	al260_20	Stone Wall	DXF, OBJ, IV
<b>Pipeline/ Pipe</b>	aq113_01	Straight Pipe Section with Supports	DXF, OBJ, IV
	aq113_02	90 Degree Pipe Section with Supports	DXF, OBJ, IV
	aq113_03	45 Degree Pipe Section with Supports	DXF, OBJ, IV
	aq113_04	Pipe Line Supports	DXF, OBJ, IV
<b>Moat</b>	bh100_01	Straight Section with Masonry Sides	DXF, OBJ, IV
	bh100_02	30 Degree Section with Masonry Sides	DXF, OBJ, IV
	bh100_03	60 Degree Section with Masonry Sides	DXF, OBJ, IV
<b>Hedgerow</b>	ea020_01	Shrub Hedgerow Section	DXF, IV



#### **A-2. Slop/Surface Configuration**

ITD Feature Category	Model Filename	Model Description	File Format
<b>Common Open Water</b>	sa010_01	Water Section	DXF, OBJ, IV

#### **A-3. Soil/Surface Materials**

ITD Feature Category	Model Filename	Model Description	File Format
<b>Ground Surface</b>	da010_01	Grass Covered Ground Section	DXF, OBJ, IV
<b>Disturbed Soil</b>	sa020_01	Soil Section	DXF, OBJ, IV
<b>Exposed Bedrock</b>	sa030_01	Bedrock Section	DXF, OBJ, IV
<b>Permanent Snowfield</b>	sa040_01	Snowfield Section	DXF, OBJ, IV

#### **A-4. Surface Drainage**

ITD Feature Category	Model Filename	Description	File format
<b>Prepared Raft/</b>	aq111_01	Ramp with Earthen Sides	DXF, OBJ, IV
<b>Bridge Site</b>	aq111_02	Ramp with Concrete Retaining Walls	DXF, OBJ, IV
	aq111_03	Ramp with Wooden Retaining Walls	DXF, OBJ, IV
	aq111_04	Ramp with Corrugated Metal Retaining Walls	DXF, OBJ, IV
<b>Canal</b>	bh020_01	Straight Earthen Canal Section	DXF, OBJ, IV
	bh020_02	15 Degree Earthen Canal Section	DXF, OBJ, IV

	bh020_03	Combined Earthen Canal Section	DXF, OBJ, IV
	bh020_04	Straight Concrete Canal Section	DXF, OBJ, IV
	bh020_05	15 Degree Concrete Canal Section	DXF, OBJ, IV
	bh020_06	Combined Concrete Canal Section	DXF, OBJ, IV
<b>Ford</b>	bh070_01	Earthen Cut Water Crossing	DXF, OBJ, IV
<b>River/Stream</b>	bh140_0A	Water Section	DXF, OBJ, IV
	bh140_01	Straight River Section with Earthen Sides	DXF, OBJ, IV
	bh140_02	15 Degree River Section with Earthen Sides	DXF, OBJ, IV
	bh140_03	Combined River Sections	DXF, OBJ, IV
<b>Dam</b>	bi020_01	Concrete Dam with Weir	DXF, OBJ, IV
	bi020_02	Wooden Dam with Weir	DXF, OBJ, IV
<b>Lock</b>	bi030_01	Canal Section with Locks	DXF, OBJ, IV
<b>Gully/Gorge</b>	DB200_01	Narrow Gully with Water	DXF, OBJ, IV
<b>Common Open Water</b>	sa010_01	Water Section	DXF, OBJ, IV
<b>Covered Drainage</b>	sa060_01	Simple Culvert Drain	DXF, OBJ, IV
	sa060_02	Simple Drain in Wall	DXF, OBJ, IV
	sa060_03	Detailed Culvert Drain	DXF, OBJ, IV
	sa060_04	Detailed Drain in Wall	DXF, OBJ, IV
	sa060_05	Simple Drain in Wall with Foundation	DXF, OBJ, IV
	sa060_06	Simple Drain in Wall with Connecting Flange	DXF, OBJ, IV
	sa060_07	Simple Culvert Drain with long Entry Walls	DXF, OBJ, IV

## A-5. Transportation

ITD Feature Category	Model Filename	Model Description	File Format
<b>RR Track</b>	an010_01	Straight Track Section with Shoulders	DXF, OBJ, IV
	an010_02	Long Radius Right Turn Track Section with Shoulders	DXF, OBJ, IV
	an010_03	Long Radius Left Turn Track Section with Shoulders	DXF, OBJ, IV
	an010_04	Short Radius Right Turn Track Section with Shoulders	DXF, OBJ, IV
	an010_05	Short Radius Left Turn Track Section with Shoulders	DXF, OBJ, IV
	an010_06	Straight Track Section	DXF, OBJ, IV
	an010_07	Long Radius Right Turn Track Section	DXF, OBJ, IV
	an010_08	Long Radius Left Turn Track Section	DXF, OBJ, IV
	an010_09	Short Radius Right Turn Track Section	DXF, OBJ, IV
	an010_10	Short Radius Left Turn Track Section	DXF, OBJ, IV
<b>Railroad Siding/Spur</b>	an050_01	Railroad Box Car	DXF, IV
	an050_02	Railroad Flat Car	DXF, IV
	an050_03	Railroad Tanker Car	DXF, IV
	an050_04	Rail Car Truck Assembly	DXF, IV
	an050_05	Railroad Engine	DXF, IV
	an050_06	Rail Yard Water Tower	DXF, IV
	an050_07	Railroad Siding Track Section with Shoulders (Left Exit)	DXF, IV
	an050_08	Railroad Siding Track Section with	DXF, IV

		Shoulders (Right Exit)	
	an050_09	Railroad Siding Track Section (Left Exit)	DXF, IV
	an050_10	Railroad Siding Track Section (Right Exit)	DXF, IV
<b>Cart Track</b>	ap010_01	Straight, Dirt Cart Path Section	DXF, OBJ, IV
	ap010_02	60 Degree and 15 Degree Combination, Dirt, cart Path Section	DXF, OBJ, IV
	ap010_03	90 Degree, Dirt, Cart Path Section	DXF, OBJ, IV
	ap010_04	Intersection, Dirt, Cart Path Section	DXF, OBJ, IV
<b>Road</b>	ap030_01	Straight, Two-Lane Road Section with Dashed Center Line, Shoulders, Curbs, and Sidewalks	DXF, OBJ, IV
	ap030_02	Stop Sign	DXF, OBJ, IV
	ap030_03	60 Degree and 15 Degree, Two-Lane Road Combination Section with Dashed Center Line, Shoulders, Curbs and Sidewalks	DXF, OBJ, IV
	ap030_04	90 Degree, Two-Lane Road Section with Dashed Center Line, Shoulders, Curbs and Sidewalks	DXF, OBJ, IV
	ap030_05	Intersection, Two-Lane Road Section with Dashed Center Line, Shoulders, Curbs, and Sidewalks	DXF, OBJ, IV
	ap030_06	Straight, Two-Lane Road Section with Solid Center Line, Shoulders, Curbs, and Sidewalks	DXF, OBJ, IV
	ap030_07	60 Degree and 15 Degree, Two-Lane Road Combination Section with Solid Center Line, Shoulders, Curbs and Sidewalks	DXF, OBJ, IV
	ap030_08	90 Degree, Two-Lane Road Section with Solid Center Line, Shoulders, Curbs and Sidewalks	DXF, OBJ, IV

	ap030_09	Intersection, Two-Lane Road Section with Solid Center Line, Shoulders, Curbs, and Sidewalks	DXF, OBJ, IV
	ap030_10	Straight, Two-Lane Road Section with Dashed Center Line	DXF, OBJ, IV
	ap030_11	60 Degree and 15 Degree, Two-Lane Road Combination Section with Dashed Center Line	DXF, OBJ, IV
	ap030_12	90 Degree, Two Lane Road Combination Section with Dashed Center Line	DXF, OBJ, IV
	ap030_13	Intersection, Two-Lane Road Combination Section with Dashed Center Line	DXF, OBJ, IV
	ap030_14	Straight, Four-Lane Divided Highway Section with Dashed Center Line	DXF, OBJ, IV
	ap030_15	60 Degree and 15 Degree, Four-Lane Divided Highway Combination Section with Dashed Center Line	DXF, OBJ, IV
	ap030_16	90 Degree, Four-Lane Divided Highway Section with Dashed Center Line	DXF, OBJ, IV
	ap030_17	Intersection, Four-Lane Divided Highway Section with Dashed Center Line	DXF, OBJ, IV
	ap030_18	Traffic Signal on Overhead Post and Arm	DXF, OBJ, IV
<b>Bridge/ Overpass/ Viaduct</b>	aq040_01	Truss Bridge with Support Structures	DXF, OBJ, IV
	aq040_02	Truss Type Suspension Bridge with Support Structures	DXF, OBJ, IV
	aq040_03	Double Span Truss Type Suspension Bridge with Support Structures	DXF, OBJ, IV
<b>Ferry Crossing</b>	aq070_01	Ramp with Wooden Ferry Docking Bumpers	DXF, OBJ, IV
<b>Sharp Curve</b>	aq118_01	90 Degree Short Radius Road Section with Dashed Center Line	DXF, OBJ, IV

<b>Tunnel</b>	aq118_02	45 Degree Short Radius Road Section with Dashed Center Line	DXF, OBJ, IV
	aq130_01	Block, Single-Lane Tunnel Entrance	DXF, OBJ, IV
	aq130_02	Extended, Block, Single-Lane Tunnel Entrance	DXF, OBJ, IV
	aq130_03	Extended, Concrete, Dual-Lane Tunnel Entrance	DXF, OBJ, IV
<b>Ford</b>	aq130_04	Concrete, Dual-Lane Tunnel Entrance with Dashed Center Line on Entrance Ramp	DXF, OBJ, IV
	bh070_01	Embankment with Earthen Sides for Water Crossing Point	DXF, OBJ, IV
<b>Runway</b>	gb055_01	Single Runway with Markings	DXF, OBJ, IV
	gb055_02	Single Runway with Markings and Taxiways	DXF, OBJ, IV
	gb055_03	Crossed Runways with Markings and Taxiways	DXF, OBJ, IV

#### A-6. Vegetation

ITD Feature Category	Model Filename	Model Description	File Format
<b>Built-up Area</b>	al020_01	Half Kilometer Area City Block	DXF
	al020_02	300 Meter Area Landscape	DXF
	al020_03	Airport Land Strip with Terrain	DXF
	al020_04	300 Meter Area Landscape	IV
<b>Barren Ground</b>	da020_01	Bare Ground Section	DXF, OBJ, IV
<b>Cropland</b>	ea010_01	Tilled Soil Section	DXF, IV
	ea010_02	Tilled Soil Section	DXF, IV
<b>Orchard/</b>	ea040_01	Orchard Section with Fruit Trees	DXF, IV

**Plantation**

<b>Vineyard</b>	ea050_01	High Detail Section of Grape Vines on Wooden Support Structure	DXF
	ea050_02	Low Detail Section of Grape Vines on Wooden Support Structures	DXF
<b>Grassland</b>	eb010_01	Grassland Section with Scrub Brush and Trees	DXF
<b>Scrub/ Brush</b>	eb020_01	Brushland Section with Scrub Brush and Trees	DXF, IV
	eb020_02	Scrub Brush with Twigs	DXF
	eb020_03	Scrub Brush	DXF
	eb020_04	Leafy Ground Cover (Simple)	DXF
	eb020_05	Leafy Ground Cover (Dense)	DXF
<b>Bamboo Cane</b>	ec010_01	Bamboo Cane in Clump	DXF
<b>Trees</b>	ec030_01	Pine Tree	DXF, OBJ, IV
	ec030_02	Christmas Tree	DXF, OBJ, IV
	ec030_03	Palm Tree	DXF, OBJ, IV
	ec030_04	Stand of Coniferous Trees	DXF, IV
	ec030_05	Tree Stump	DXF, OBJ, IV
	ec030_06	Felled Dead Tree with Stump	DXF, OBJ, IV
	ec030_07	Dead Tree, Standing	DXF, OBJ, IV
	ec030_08	Tree	DXF, OBJ, IV
	ec030_09	Tree	DXF, IV
	ec030_10	Tree	DXF, IV
	ec030_11	Tree	DXF, IV
	ec030_12	Tree	DXF

ec030_13	Tree	DXF, IV
ec030_14	Tree	DXF, IV
ec030_15	Tree	DXF, IV
ec030_16	Tree	DXF, IV
ec030_17	Tree	DXF, IV
ec030_18	Coniferous Tree	DXF, OBJ, IV



## APPENDIX B

### DATABASE TABLES AND DATA ELEMENTS

#### B-1. model\_info

<u>Data Elements</u>	<u>Description</u>
model num	Model number (serial key)
contry cd	Country code
service	Service name
branch	Branch name
category	Model major category
model nm	Model name
model desc	Model description
model type	Model type (STATIC/DYNAMIC)
sub cat	Model sub-category
model id	Model ID (FACC code for ITD models)
version	Version code
length	Model length (dimension in longitude direction)
width	Model width (dimension in latitude direction)
height	Model height (dimension in altitude direction)
unit	Unit of measure (e.g. meters)
polygon num	Number of polygons
file nm	Model path and file name
file format	Model file format (e.g. DXF, OBJ etc.)
dist txt	Distribution code
class txt	Classification
proponent	Organization owns the model
poc	Point of contact name
poc phone	Point of contact phone number
creator	Company/Name who drew the model
creation dt	Date the model was created
comments	Comments about the model
spec	Government specification
fsc num	Federal supply class number
draw num	Drawing number
acc ind	Accessory indicator (Y=Yes, N=No)

area	Area
option6	Not currently used
option7	Not currently used
option8	Not currently used
option9	Not currently used
option10	Not currently used

## **B-2. country**

<u>Data Elements</u>	<u>Description</u>
cntry num	Country number (serial key)
cntry cd	Country code
cntry nm	Country name

## **B-3. service**

<u>Data Elements</u>	<u>Description</u>
serv num	Service number (serial key)
serv nm	Service name

## **B-4. branch**

<u>Data Elements</u>	<u>Description</u>
br num	Branch number (serial key)
branch nm	Branch name
cntry cd	Country code (foreign key to country table)
serv nm	Service name (foreign key to service table)

**B-5. category**

<u>Data Elements</u>	<u>Description</u>
cat num	Category number (serial key)
cat nm	Category name
cntry cd	Country code (foreign key to country table)
serv nm	Service name (foreign key to service table)
branch nm	Branch name (foreign key to branch table)

**B-6. distribution**

<u>Data Elements</u>	<u>Description</u>
dist cd	Distribution code
dist desc	Distribution description

## APPENDIX C

### AN EXAMPLE OF USING Ez3d TO ADD TEXTURE

The example presented in this appendix is a detailed walk through of the procedures we used to add texture to a DXF model, and to convert the model to OBJ format. A simple model (a horizontal slat fence) is used for the demonstration. Problems (1) - (4) mentioned in Section 4.3.2 were covered in the example.

#### Example:

The DXF model of a horizontal slat fence is to be partially textured and converted to OBJ format. The model is located in `/TEC/models/ITD/DXF` directory and named as `al260_02.dxf`. Use the following steps to complete the task.

- (1) From your home directory, type:

**Ez3d**

to start Ez3d software.

- (2) In "Project Selector" window, under **Select from "project" or enter new Project name**, append "example.ez3d" to the existing text. Press **OK**.
- (3) From another UNIX shell, type the following commands:

```
cd ~/Ez3d_projects/example.ez3d/scene
```

```
cp /TEC/models/ITD/DXF/al260_02.dxf .
```

This step copies the DXF model into the project directory.

- (4) Choose **Sculptor - Load** option from the main menu. In the "Sculptor Load" window, select "al260\_02.dxf" from the file list and press **OK**.
- (5) In "Please confirm" window, press **Ok**.
- (6) In the "Sculptor Load" window, press **Cancel** to close it.
- (7) Choose **"Costume - Display - Shaded"** option from the main menu to render the model.
- (8) Unselect the model.
- (9) Choose **"Costume - Texture"** to open the "Texture Editor" window.
- (10) Choose **"Sculptor - Object List"** option to open the "Object List" window. In this window a list of model components is displayed:

al260\_02  
al260\_02\_MODEL  
al260\_02\_MODEL\_1  
al260\_02\_MODEL\_2  
al260\_02\_MODEL\_3

In above list, the first object is the entire model, the other four are the four components of the model. Each component cannot be broken down further. Texture can be put on each model component. Select one of the three horizontal bars by highlighting **al260\_02\_MODEL**.

- (11) Go to the "Texture Editor" window, choose **"Palette - System Palette - surfaces"** from the menu. Turn on the "Automatic Update" flag.
- (12) Choose **wood.4.rgb** from the "Texture Editor" window. You can see that texture has been added to the middle bar of the fence.
- (13) Go to the "Object List" window to highlight **al260\_02\_MODEL\_3**. Another horizontal bar (the top one) is selected. Choose the **wood.4.rgb** image from the "Texture Editor" window to add texture to it.
- (14) In the "Object List" window, both **al260\_02\_MODEL** and **al260\_02\_MODEL\_3** should be highlighted by now. If not, be sure they are.
- (15) Choose **"Edit - Group"** option from the main menu to group the textured parts. A new object **Group001** appears in the "Object List" window. Unhighlight this object.
- (16) In the "Object List" window, highlight **al260\_02\_MODEL\_1** and **al260\_02\_MODEL\_2**.
- (17) Choose **"Edit - Group"** option from the main menu to group the non-textured parts. A new object **Group002** appears in the "Object List" window. Unhighlight this object.
- (18) Choose **"Edit - Select All"** option from the main menu to select entire model.
- (19) To center and rotate the model: choose **"Sculptor - Slider Interactors"** option from main menu to open the "Slider Interactors" window. From this window, select the **CENTER** option, and change all three numbers to **0.00**. You must press Enter after each entry. This will center the model. Again from this window, select the **ROTATIONS** option, and change the first number from **0.00** to **90.0**, then press Enter. This will rotate the model 90 degrees around the x axis.
- (20) Close the "Slider Interactors" window.
- (21) Choose **"Sculptor - Save"** from the main menu to open the "sculptor Save" window, and select **WAVEFRONT.obj** under the File format. Press **OK** to save the model in OBJ format.

- (22) Press **Ok** in the “For your information” window.
- (23) Choose “**File - Exit**” from the main menu to exit Ez3d.
- (24) Select **Don't Save** in the “Please confirm” window.
- (25) Go to the project directory by typing:

```
cd ~/Ez3d_projects/example.ez3d/scene
```

- (26) Do a file list, you should see the following three files:

```
al260_02.object.mtl
al260_02.object.obj
wood.4.rgb
```

- (27) These files were generated by Ez3d. The first one is the material library file that defines the colors and textures used by the model. The second file is the OBJ file that defines the model geometry. The third one is a symbolic link to the texture map.
- (28) Use a text editor to view the **al260\_02.object.obj** file. It contains four groups separated by “**usemtl Material\_x**”. Each group defines one model component on the “Object List” inside Ez3d. You can see that only the first two groups have the corresponding texture vertices (indicated by **vt**), therefore they are the two textured horizontal bars. The corresponding materials for them are **Material\_1** and **Material\_2**. You need to remember this in order to perform the next step.
- (29) Edit the material file: **al260\_02.object.mtl**, locate the definitions for **Material\_1** and **Material\_2**, and insert the following line under the definitions:

```
map_Kd wood.4.rgb
```

The modified file should look like:

```
# Generated by Radiance Software's Ez3d Modeler
#      Fri Dec 29 06:07:00 1995
# Ez3d is a registered copyright of Radiance Software International.
#
```

```
newmtl Material_1
  map_Kd wood.4.rgb
  Ka 0.167500 0.125625 0.000000
  Kd 0.670000 0.502500 0.000000
  Ks 0.200000 0.200000 0.200000
  Ni 1.000000
```

```
newmtl Material_2
  map_Kd wood.4.rgb
```

Ka 0.167500 0.125625 0.000000  
Kd 0.670000 0.502500 0.000000  
Ks 0.200000 0.200000 0.200000  
Ni 1.000000

newmtl Material\_3

Ka 0.167500 0.125625 0.000000  
Kd 0.670000 0.502500 0.000000  
Ks 0.200000 0.200000 0.200000  
Ni 1.000000

newmtl Material\_4

Ka 0.167500 0.125625 0.000000  
Kd 0.670000 0.502500 0.000000  
Ks 0.200000 0.200000 0.200000  
Ni 1.000000

(30) View the resulting OBJ model using the Performer viewer **perfly**:

**perfly al260\_02.object.obj**

Apply texture to see that the texture was mapped onto two horizontal bars, while other parts of the fence are non-textured.